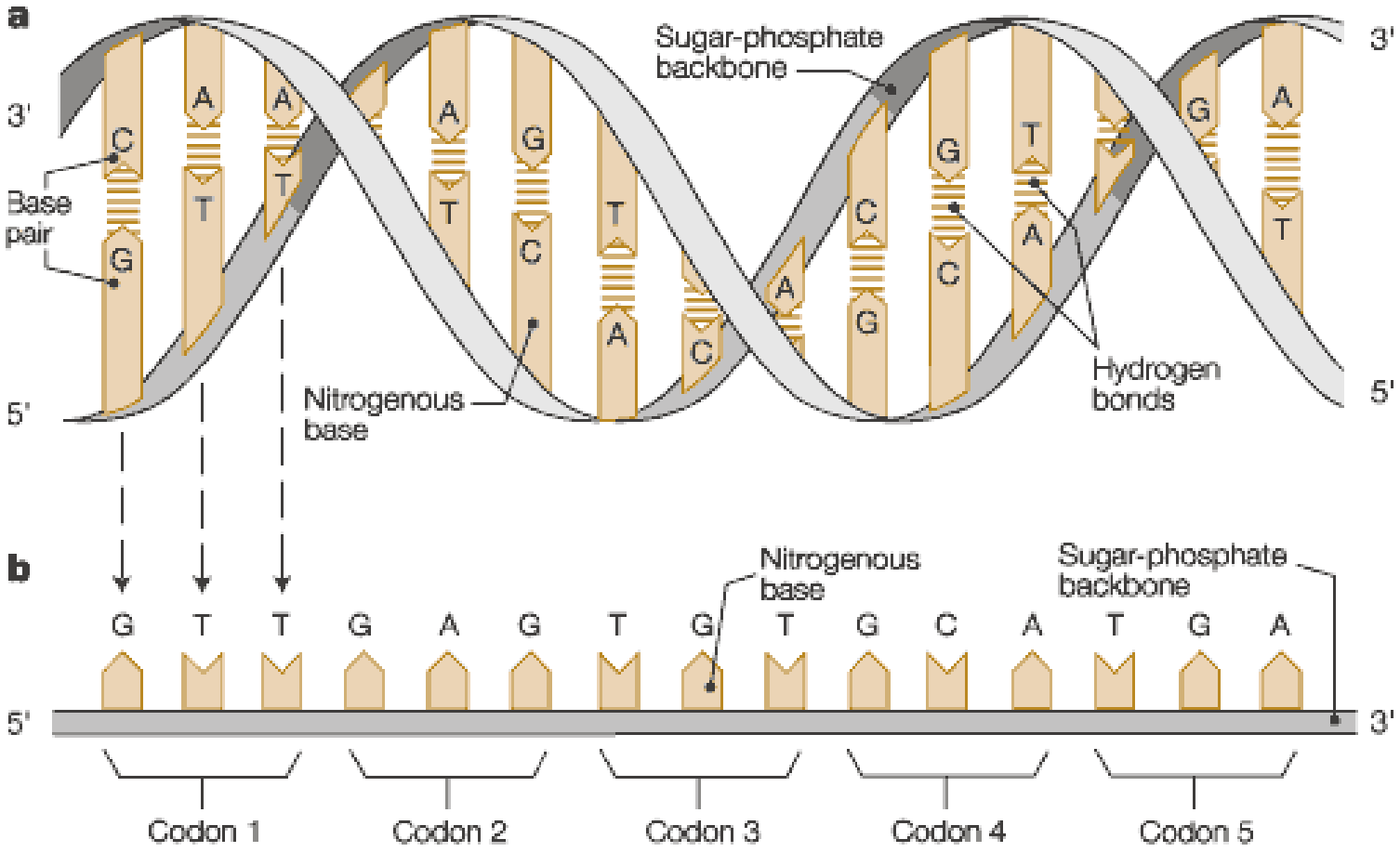
New Architectures for a New Biology

Martin M. Deneroff
deneroff@deshaw.com

D. E. Shaw Research, LLC

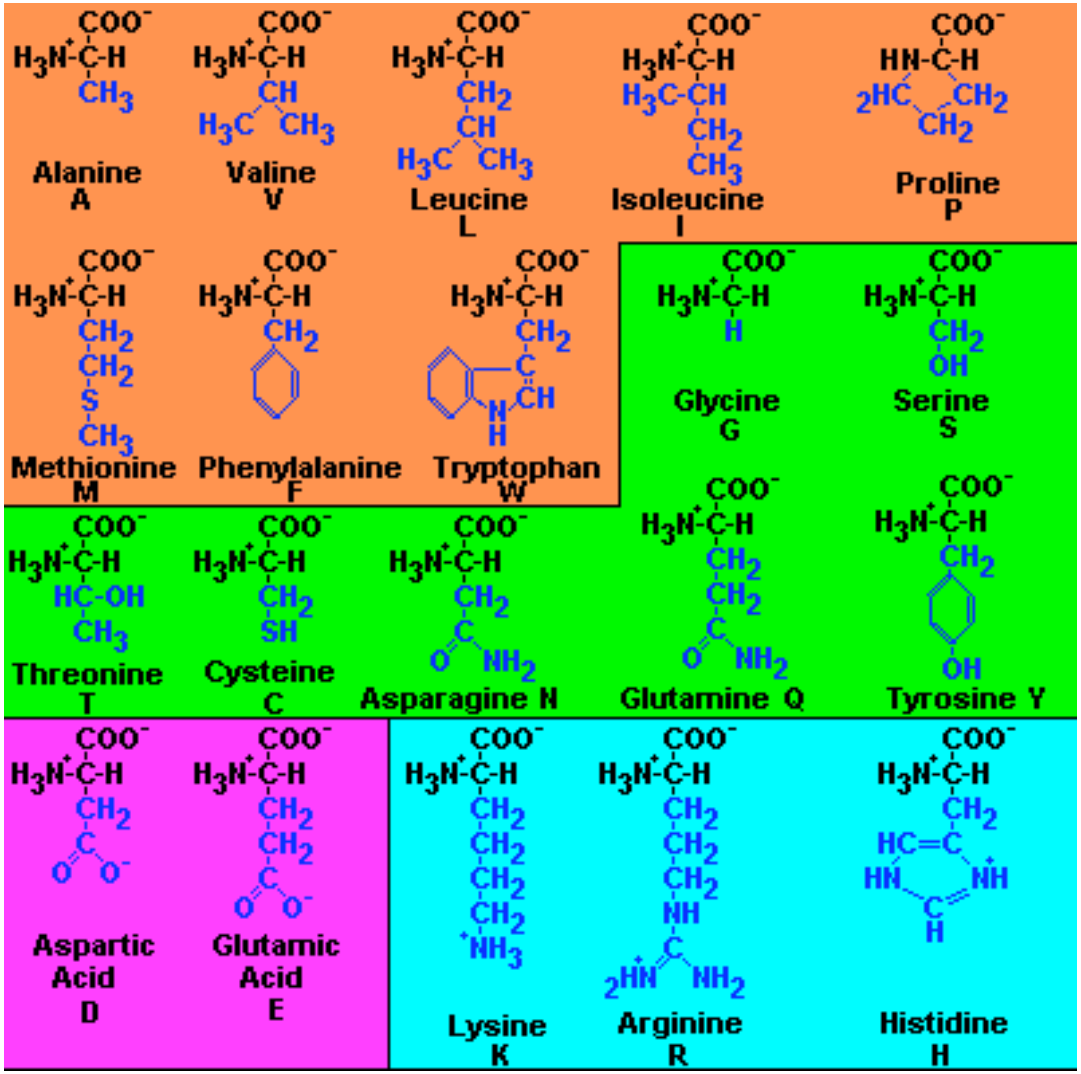
Background
(A Bit of Basic Biochemistry)

DNA Codes for Proteins

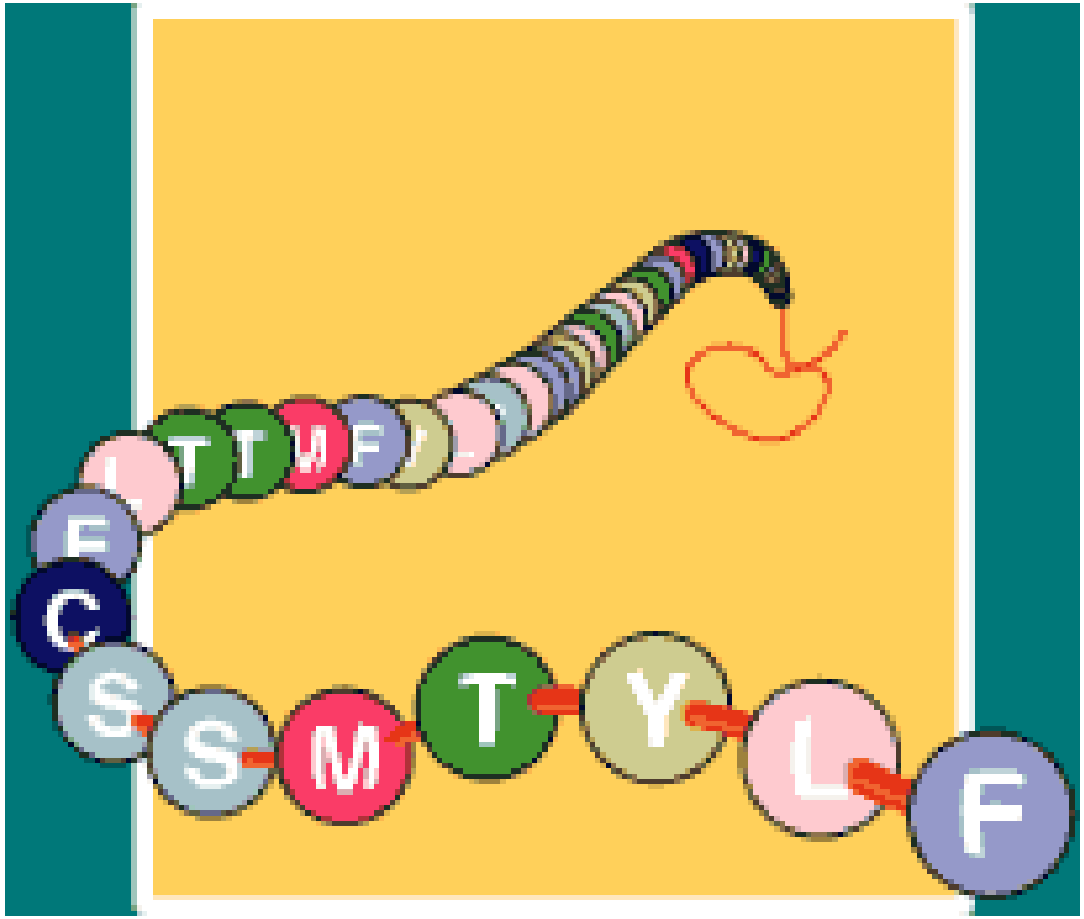


Source: *Molecular Biology of the Cell*, Garland Publishing, Inc., New York, 1994

The 20 Amino Acids

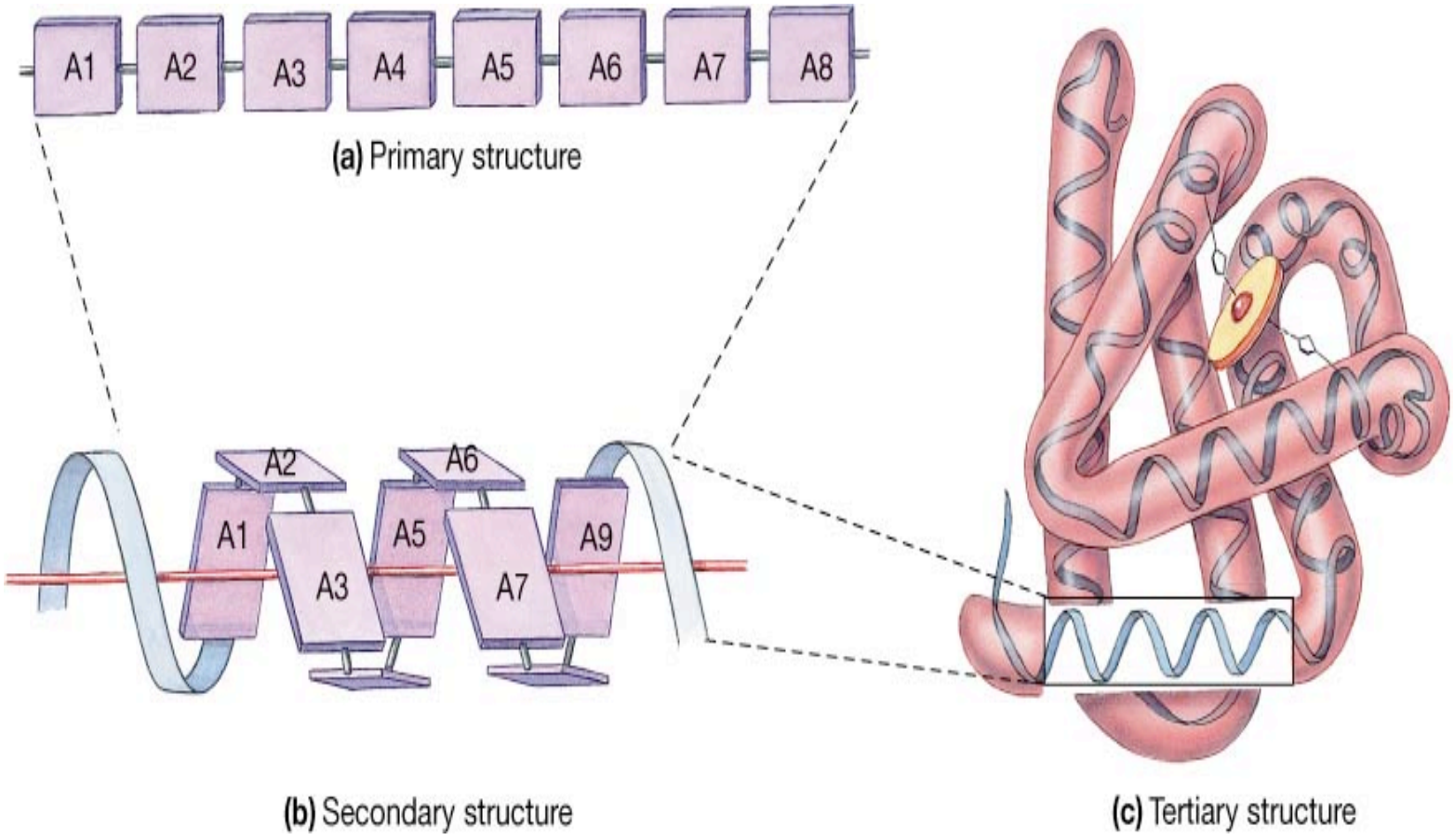


Polypeptide Chain



Source: www.yourgenome.org

Levels of Protein Structure



Source: Robert Melamede, U. Colorado

What We Know and What We Don’t

- Decoded the genome
- Don’t know most protein structures
  - Especially membrane proteins
- No detailed picture of what most proteins do
- Don’t know how everything fits together into a working system

We Now Have The Parts List ...

SECTION 7 PARTS LIST

7.1 BOARD A-I

REFERENCE	ARP PART NUMBER	ARP/MFG NUMBER
A1, 2	5601801	A-2801-008A/SL 19988
A3, 4	5601901	A-2801-009/SL19986
Q6,8	1303901	IMF3958
Q2,3	1301701	2N5172
Q9	5600401	A2802-014-1
Q1,4,5	1302801	2N6076
CR1-6	1200301	IN4148
P2	1001203	B2801-006-1B
P1	5701801	B-2801-010-1A
T3,T4	1000903	U201R251B
T1	1000904	R201R102B
T2	1000913	U201R103B
C8	1100901	WCR1P47
C2	1100702	150D406X9010B
C4,5	1100608	G-0-001-G-10-0
S1	1900801	02-481-0001
S2	1902401	02-481-0006

7.2 BOARD B-I

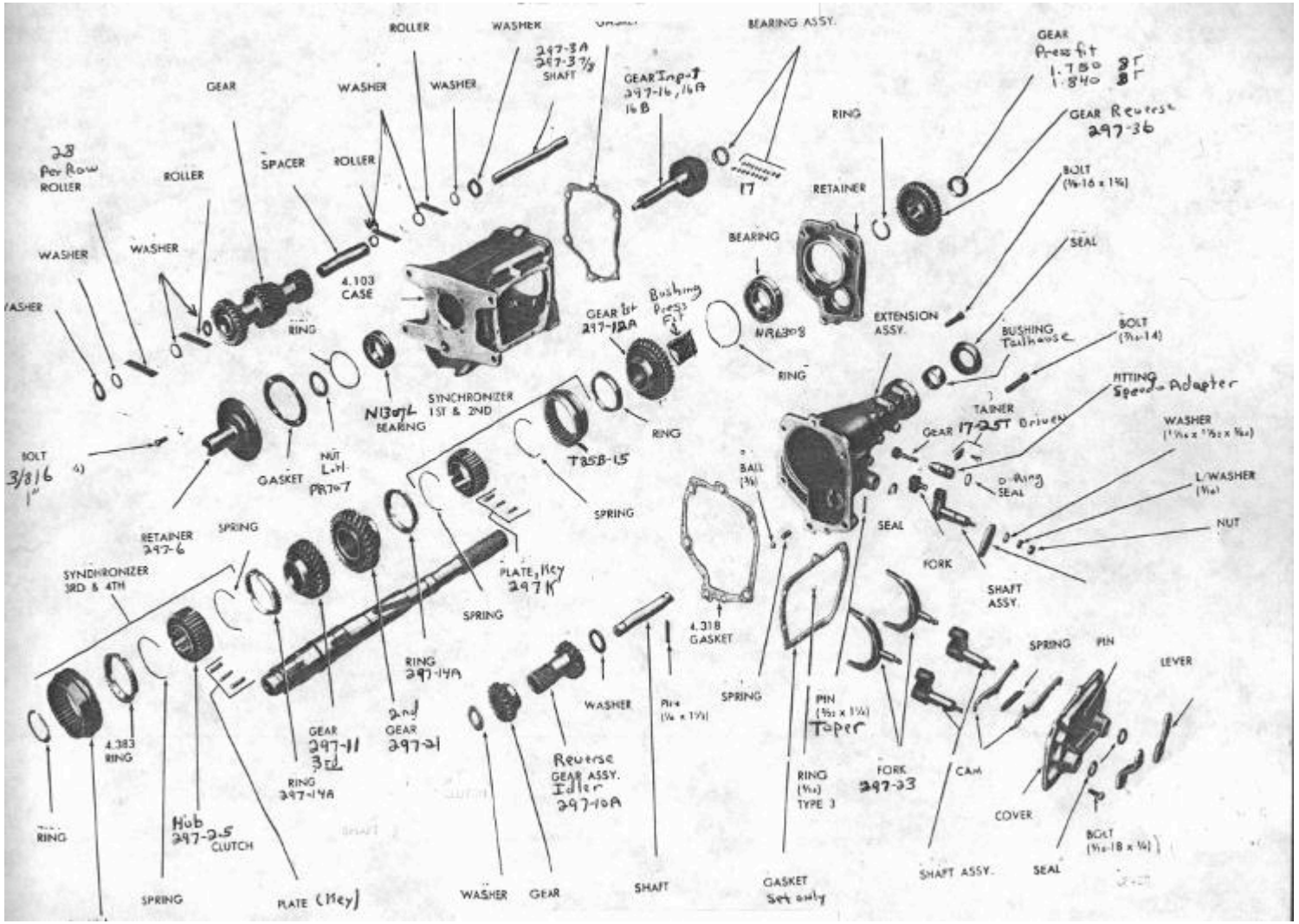
REFERENCE	ARP PART NUMBER	ARP/MFG NUMBER
A1,2,	5601801	A2801-008A
A3	5601501	A4024-006-2B
Z1,2	5602001	A2803-002A
Q1,9,10,16,17,18	1301701	2N5172
Q7,14	1302801	2N6076
Q4/5,Q11/12	7502600	APL4027-008
Q2,8,15	1302501	2N5461
Q3	5600201	A2803-003-1B
Q6,13	5600202	A2803-003-2B
CR1-3,5-12	1200301	1N4148
CR4	1200102	1N34
C12,16	1101201	DM-15-681K
C10,11	1100612	Tag-00-10/35-50/20
R32,44	1000105	SA-21
P16	5700701	B2801-006-1D
P5,6,7,10,11,	5700702	B2801-006-2B
P1,2,3,4,8,9,12,		
13,14,15	5700703	B2801-006-3B
T1,4,7	1000909	U201R103B
T2,3,5,6	1000915	U201R104B
S1-11	1902401	01-481-0006

7.3 BOARD C-I

REFERENCE	ARP PART NUMBER	ARP/MFG NUMBER
M1	4023	
A1	5601901	A-2801-009-1
A2	5601501	B4023-006-2B
Q12	1304601	TZ81
Q2,3,4,6.8,		
10,13,16,18	1301701	2N5172
Q2,5,7,9,11,		
14,15,17	1302801	2N6076
CR1-22	1200301	1N4148
C7,8	1100602	TAG-00-3.3/20-10/10

But We Don’t Know What the Parts Look Like …

Or How They Fit Together ...



Or How The Whole Machine Works

How Can We Get There?

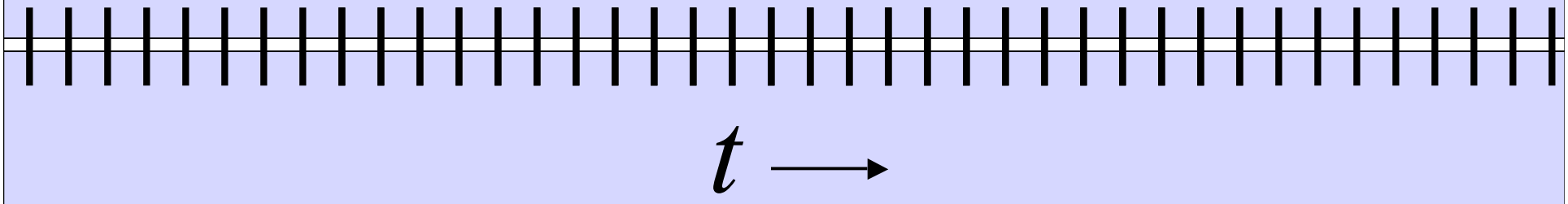
Two major approaches:

- Experiments
  - _ Wet lab
  - _ Hard, since everything is so small
- Simulation
  - _ Simulate:
    - · How proteins fold (structure, dynamics)
    - · How proteins interact with
      - - Other proteins
      - - Nucleic acids
      - - Drug molecules
  - _ Gold standard: Molecular dynamics (MD)

Molecular Dynamics

Molecular Dynamics

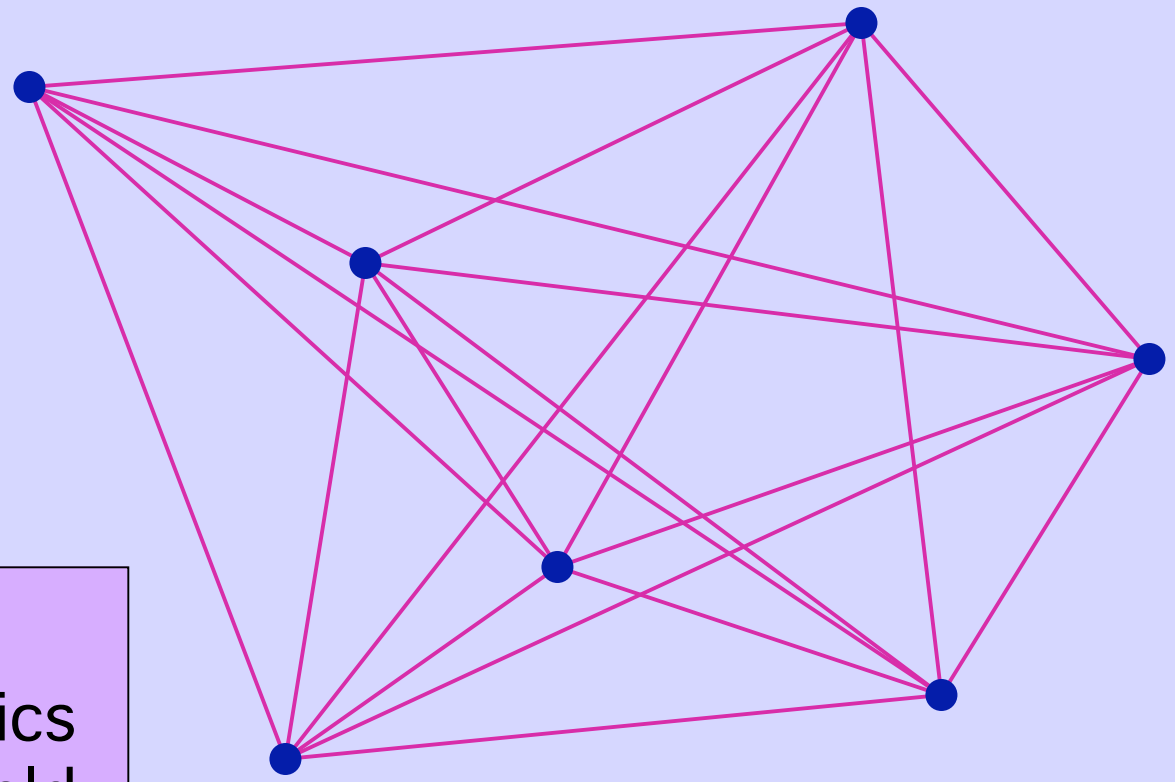
Divide time into discrete time steps



~1 fs time step

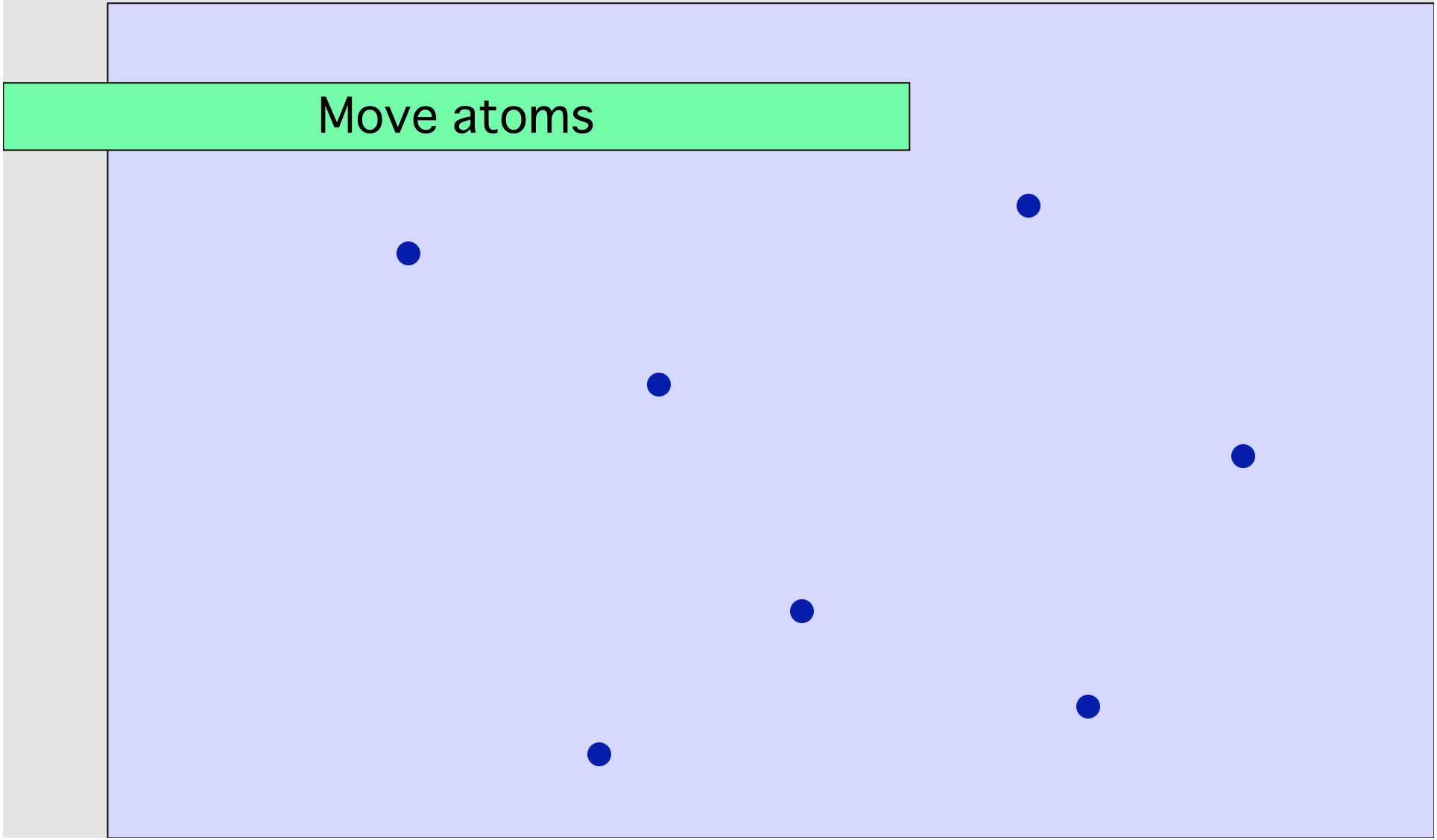
Molecular Dynamics

Calculate forces

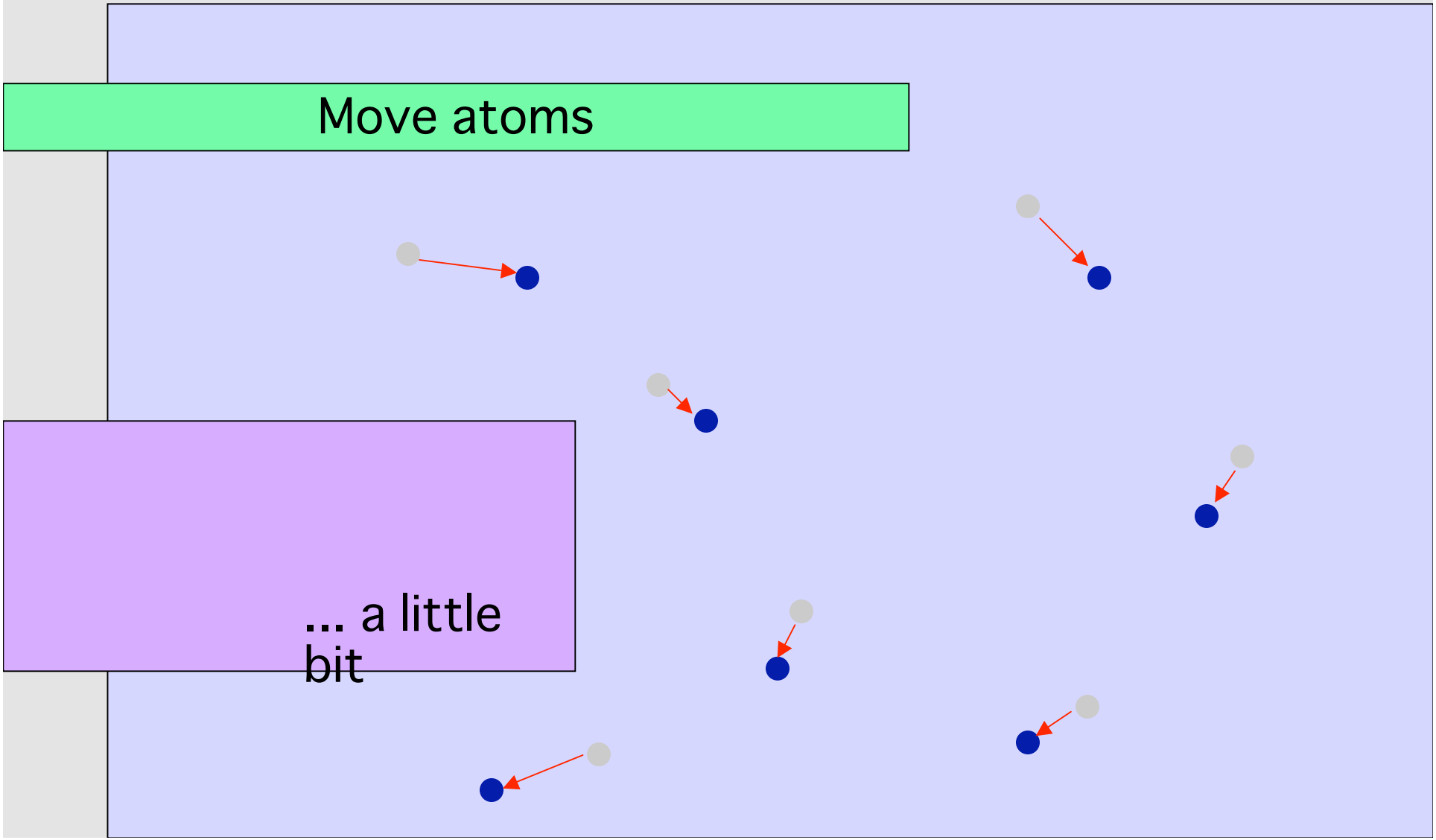


Molecular mechanics force field

Molecular Dynamics



Molecular Dynamics



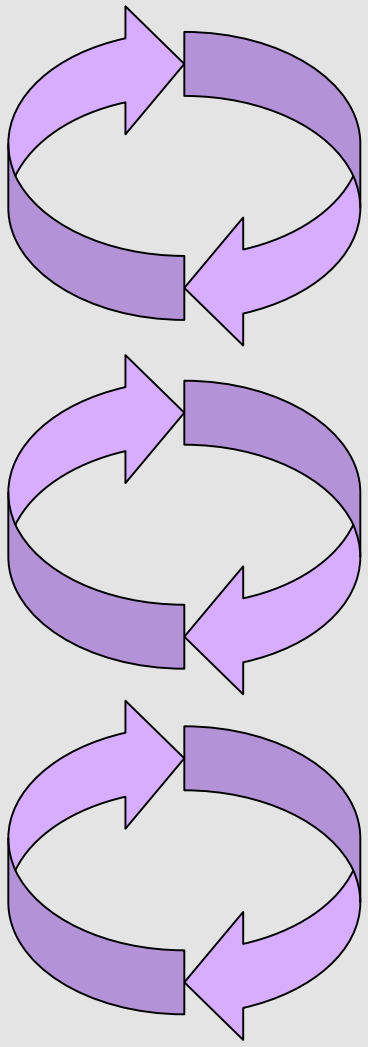
Molecular Dynamics

Iterate

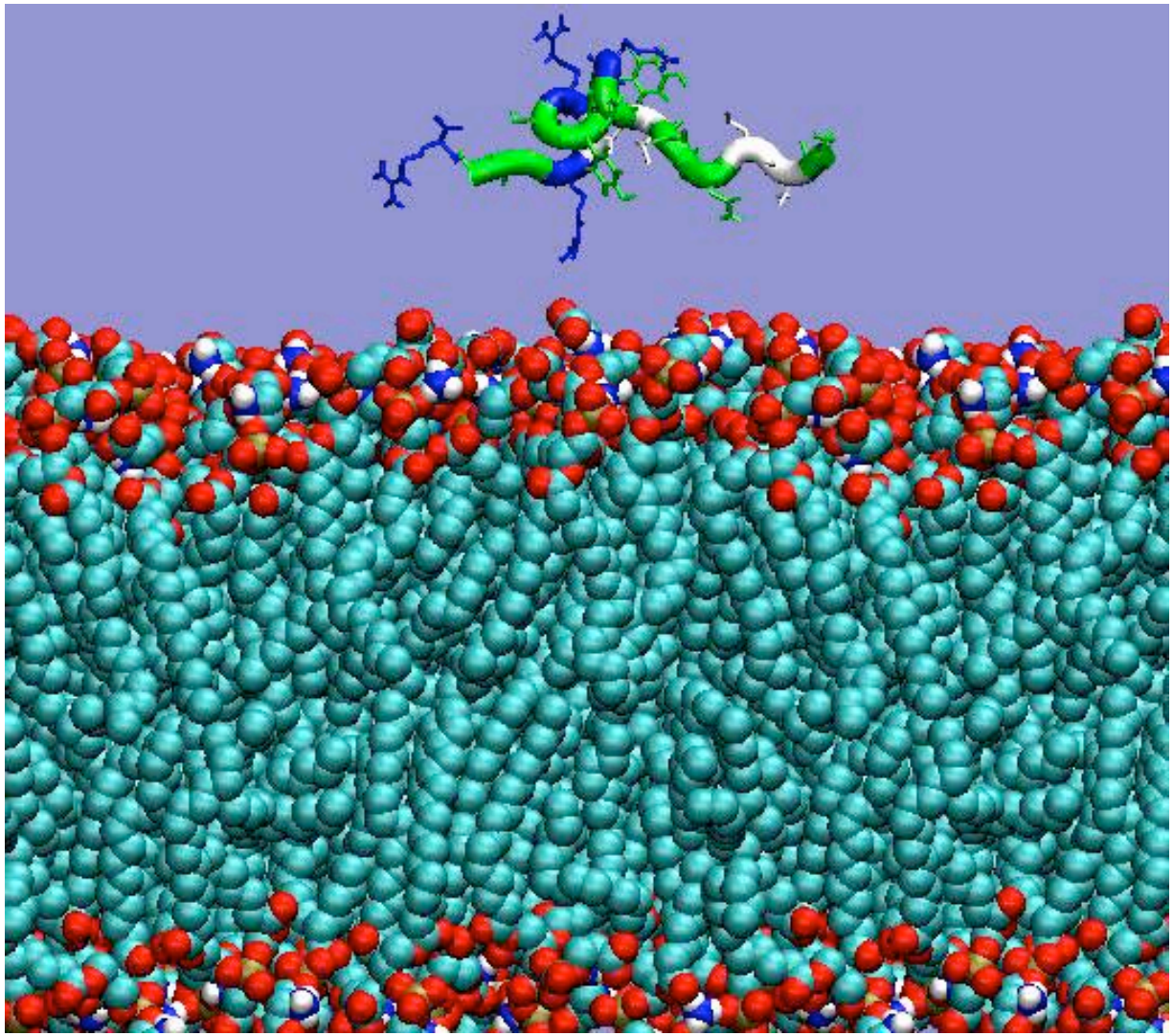
... and iterate

... and iterate

Integrate Newton's laws of motion



Example of an MD Simulation



Main Problem With MD

Too slow!

Example I just showed:

- 2 ns simulated time
- 3.4 CPU-days to simulate

Goals and Strategy

Thought Experiment

- What if MD were
  - _ Perfectly accurate?
  - _ Infinitely fast?

- Would be easy to perform
  arbitrary computational experiments
  - _ Determine structures by watching them form
  - _ Figure out what happens by watching it happen
  - _ Transform measurement into data mining

Two Distinct Problems

Problem 1: Simulate many short trajectories

Problem 2: Simulate one long trajectory

Simulating Many Short Trajectories

- Can answer surprising number of interesting questions
- Can be done using
  - Many slow computers
  - Distributed processing approach
  - Little inter-processor communication
- E.g., Pande's *Folding at Home* project

Simulating One Long Trajectory

- Harder problem
- Essential to elucidate many biologically interesting processes
- Requires a single machine with
  - Extremely high performance
  - Truly massive parallelism
  - Lots of inter-processor communication

DESRES Goal

- Single, millisecond-scale MD simulations (long trajectories)
  - Protein with 64K or more atoms
  - Explicit water molecules
- Why?
  - That's the time scale at which many biologically interesting things start to happen

Protein Folding

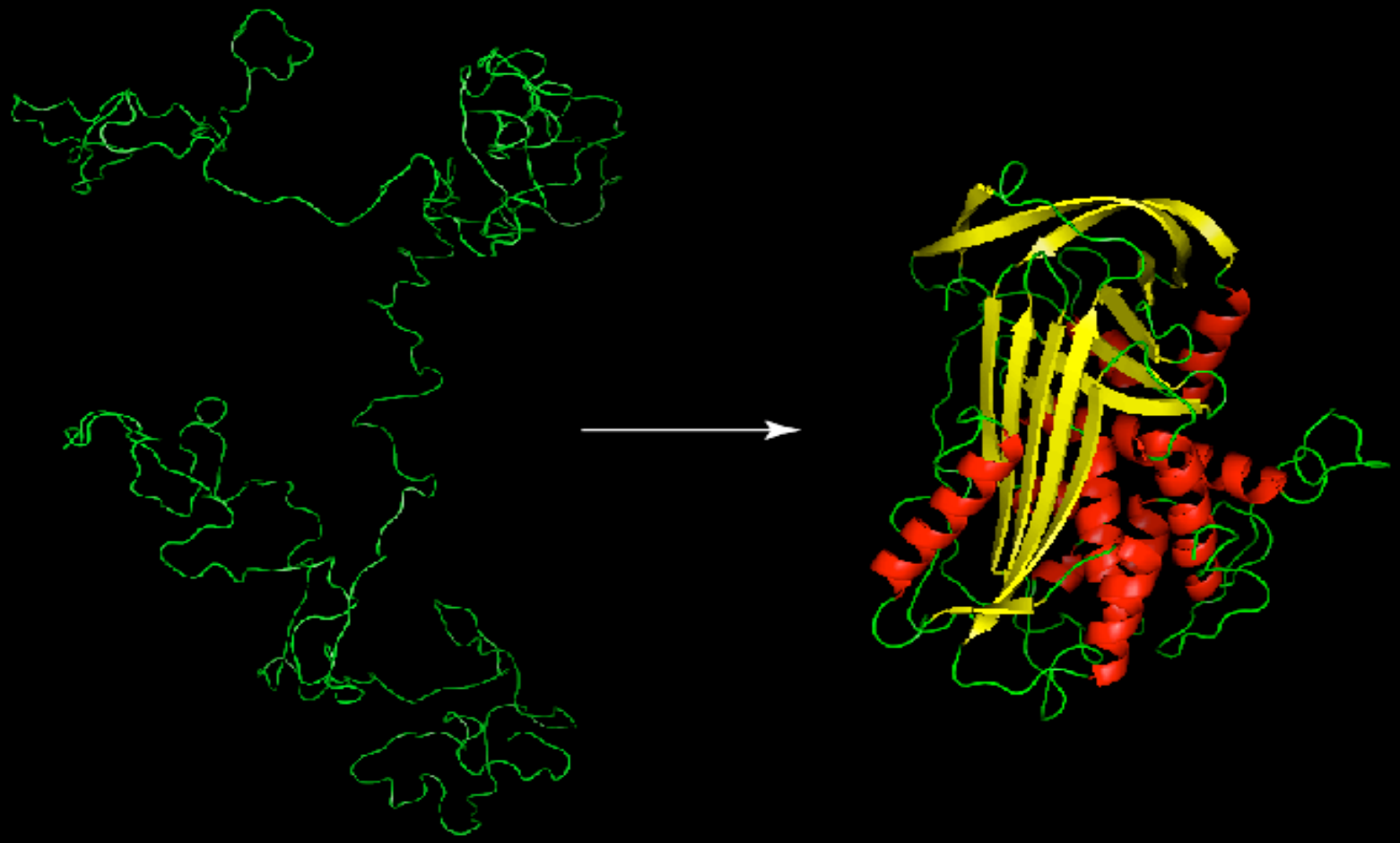


Image: Istvan Kolossvary & Annabel Todd,
D. E. Shaw Research

Interactions Between Proteins

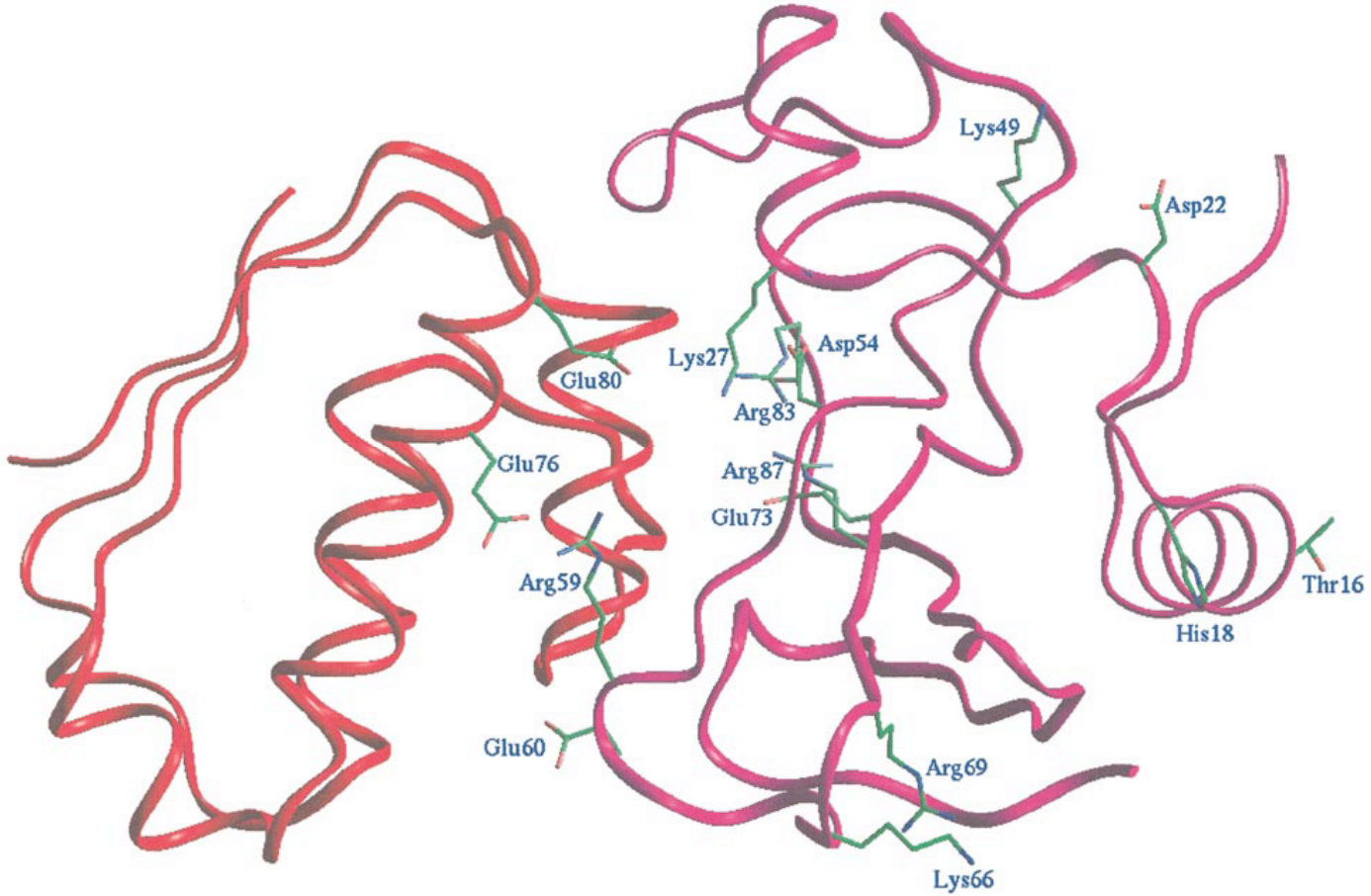
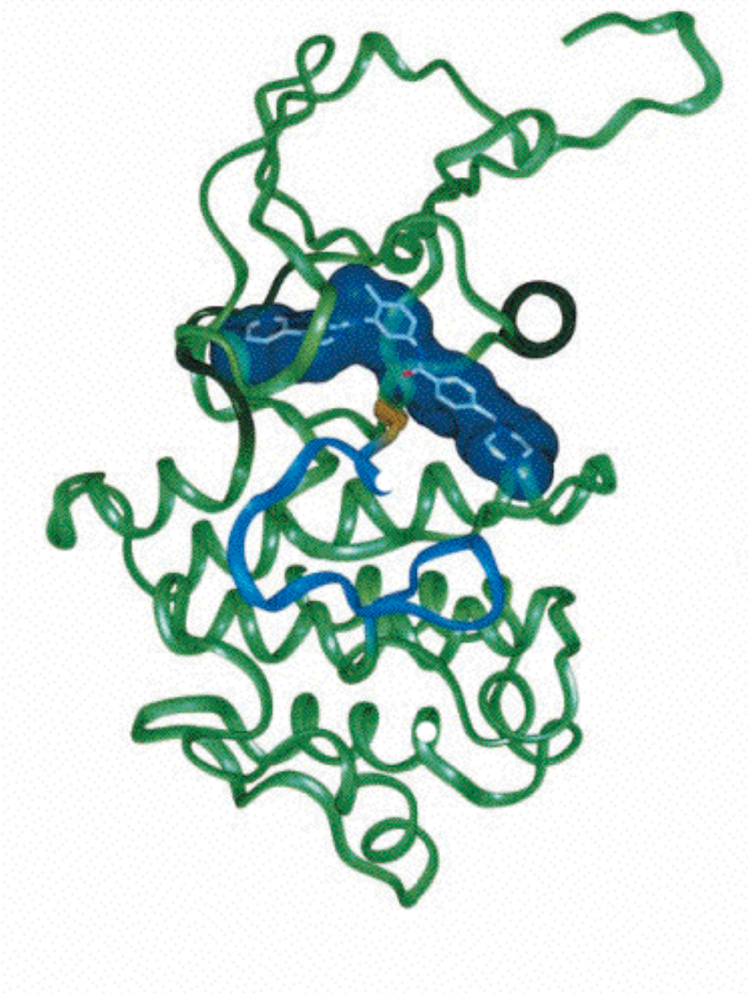


Image: Vijayakumar, et al., *J. Mol. Biol.* 278, 1015 (1998)

Binding of Drugs to their Molecular Targets



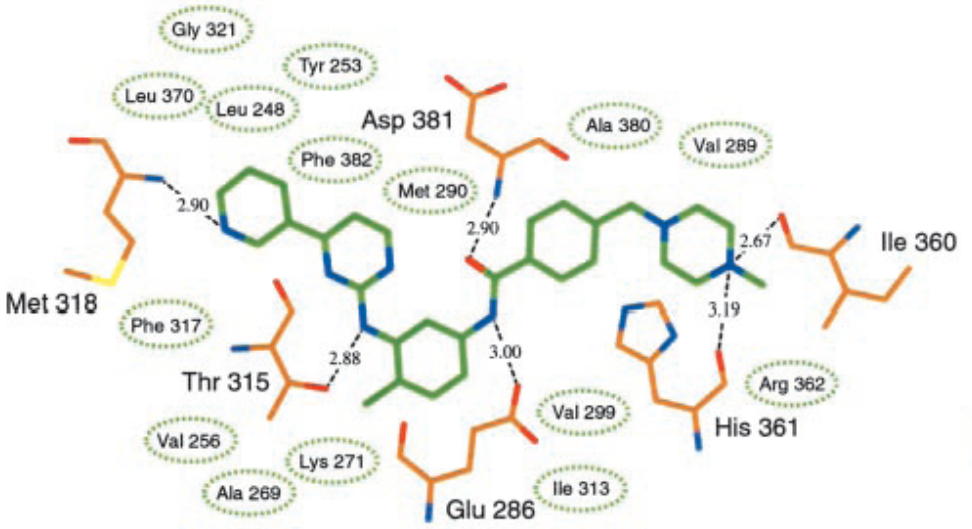
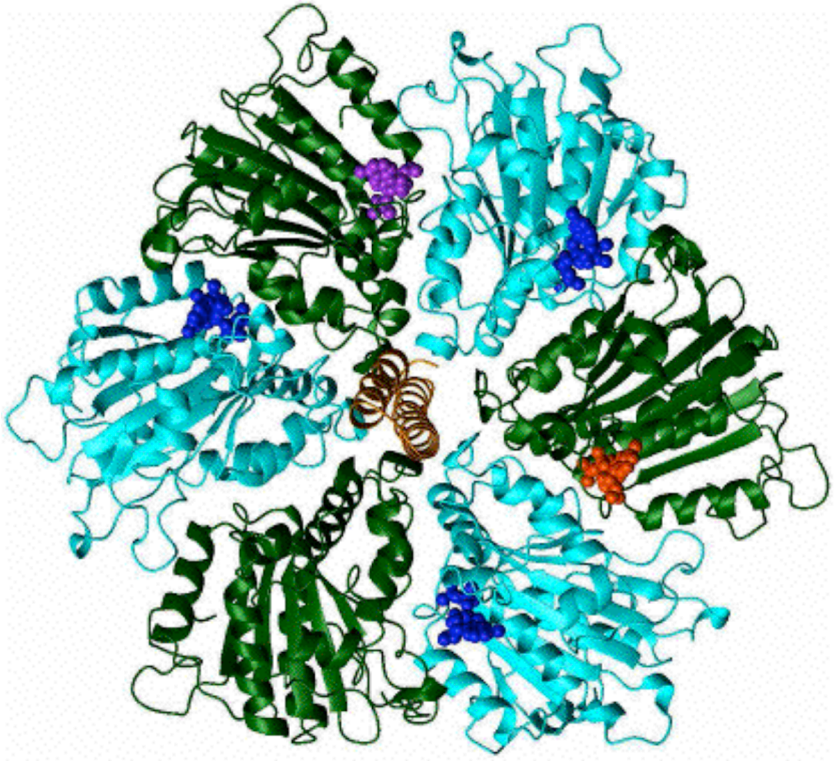


Image: Nagar, et al., *Cancer Res.* 62, 4236 (2002)

Mechanisms of Intracellular Machines



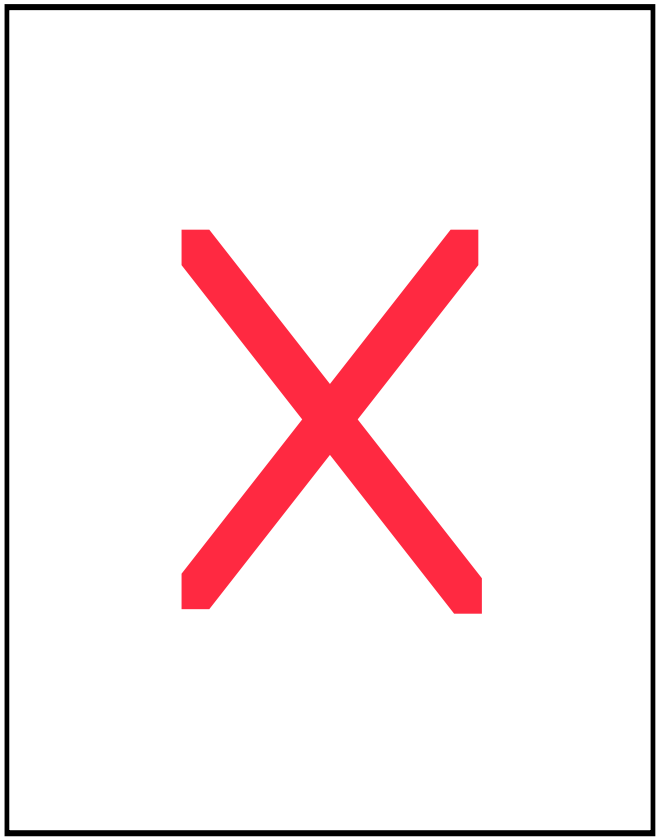
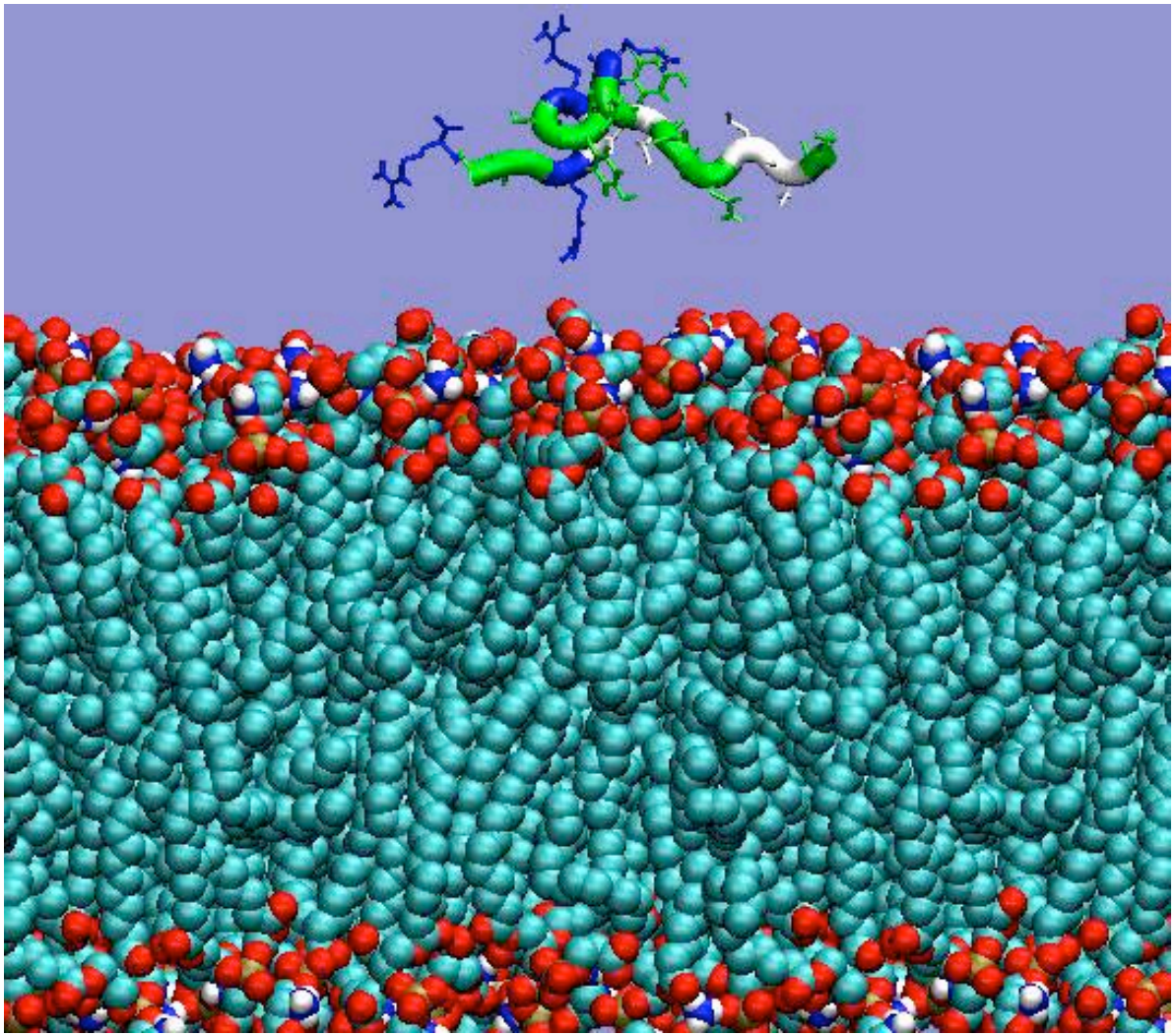


Image: H. Grubm_ller, in Attig, et al. (eds.), *Computational Soft Matter* (2004)

What Will It Take to Simulate a Millisecond?

- We need an enormous increase in speed
  - Current (single processor): ~ 100 ms / fs
  - Goal will require < 10 μs / fs
- Required speedup:
  - > 10,000x faster than current single-processor speed
  - ~ 1,000x faster than current parallel implementations
- *Can't accept >10,000x the power (~5 Megawatts)!*

Target Simulation Speed



3.4 days today
(one processor)

~ 13 seconds on
our machine
(one segment)

Molecular Mechanics Force Field

Stretch

Bend

Torsion

Bonded

Electrostatic

Van der Waals

Non-Bonded

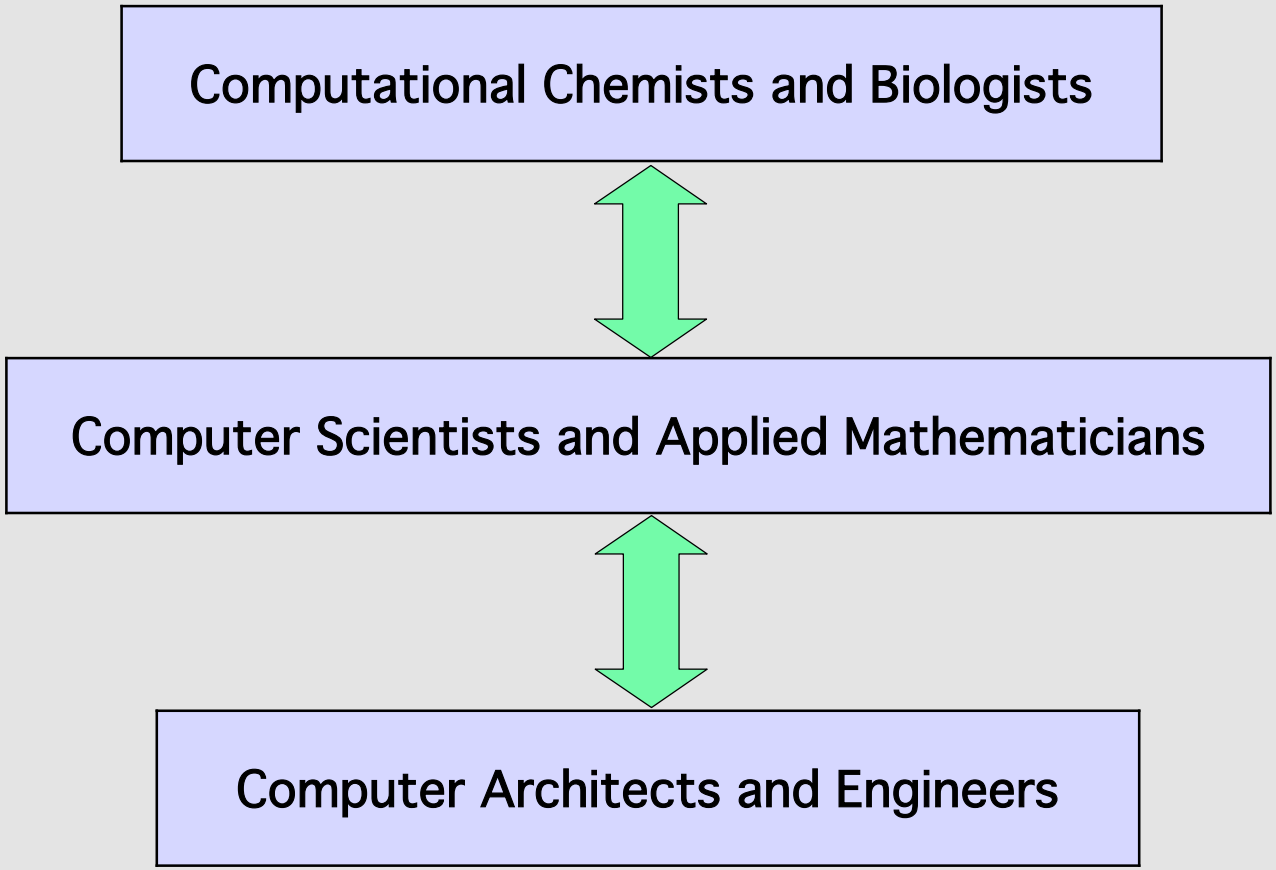
What Takes So Long?

- Inner loop of force field evaluation looks at all *pairs* of atoms (within distance *R*)
- On the order of 64K atoms in typical system
- Repeat ~10^{12} times
- Current approaches too slow by several orders of magnitude
- What can be done?

Our Strategy

- New architectures
  - Design a specialized machine
  - Enormously parallel architecture
  - Based on special-purpose ASICs
  - Dramatically faster for MD, but less flexible
  - Projected completion: 2008
- New algorithms
  - Applicable to
    - Conventional clusters
    - Our own machine
  - Scale to very large # of processing elements

Interdisciplinary Lab



New Architectures

Alternative Machine Architectures

- Conventional cluster of commodity processors
- General-purpose scientific supercomputer
- Special-purpose molecular dynamics machine

Conventional Cluster of Commodity Processors

- Strengths:
  - _ Flexibility
  - _ Mass market economies of scale
- Limitations
  - _ Doesn’t exploit special features of the problem
  - _ Communication bottlenecks
    - · Between processor and memory
    - · Among processors
  - _ Insufficient arithmetic power

General-Purpose Scientific Supercomputer

- E.g., IBM *Blue Gene*
- More demanding goal than ours
  - General-purpose scientific supercomputing
  - Fast for wide range of applications
- Strengths:
  - Flexibility
  - Ease of programmability
- Limitations for MD simulations
  - Expensive
  - Still not fast enough for our purposes

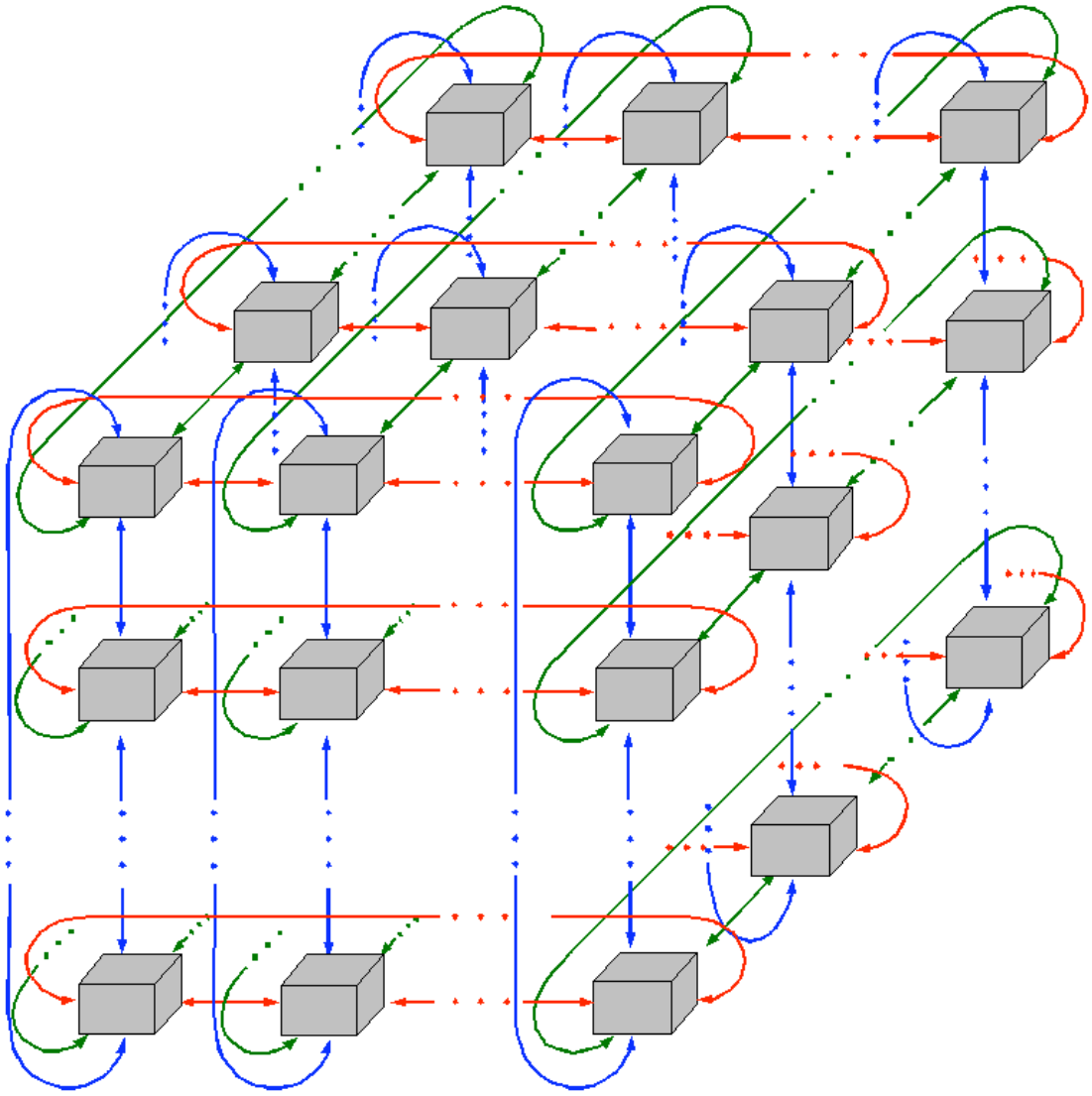
Anton: DESRES' Special-Purpose MD Machine

- Strengths:
  - Several orders of magnitude faster for MD
  - Excellent cost/performance characteristics
- Limitations:
  - Not designed for other scientific applications
    - They'd be difficult to program
    - Still wouldn't be especially fast
  - Limited flexibility

Anton System-Level Organization

- Multiple segments (probably 8 in first machine)
- 512 nodes (each consists of one ASIC plus DRAM) per segment
  - Organized in an 8 x 8 x 8 toroidal mesh
- Each ASIC equivalent performance to roughly 500 general purpose microprocessors
  - ASIC power similar to a single microprocessor

3D Torus Network



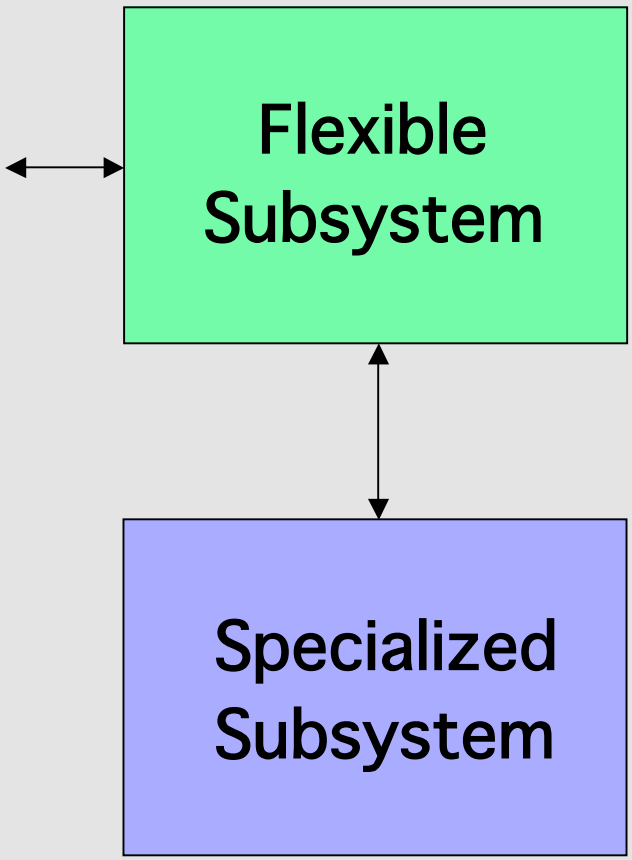
Why a 3D Torus?

- Topology reflects physical space being simulated:
  - Three-dimensional nearest neighbor connections
  - Periodic boundary conditions
- Bulk of communications is to near neighbors
  - No switching to reach immediate neighbors

Source of Speedup on Our Machine

- Judicious use of arithmetic specialization
  - Flexibility, programmability only where needed
  - Elsewhere, hardware tailored for speed
    - Tables and parameters, but not programmable
- Carefully choreographed communication
  - Data flows to just where it's needed
  - Almost never need to access off-chip memory

Two Subsystems on Each ASIC



- Programmable, general-purpose
- Efficient geometric operations
- Modest clock rate

- Pairwise point interactions
- Enormously parallel
- Aggressive clock rate

Where We Use Specialized Hardware

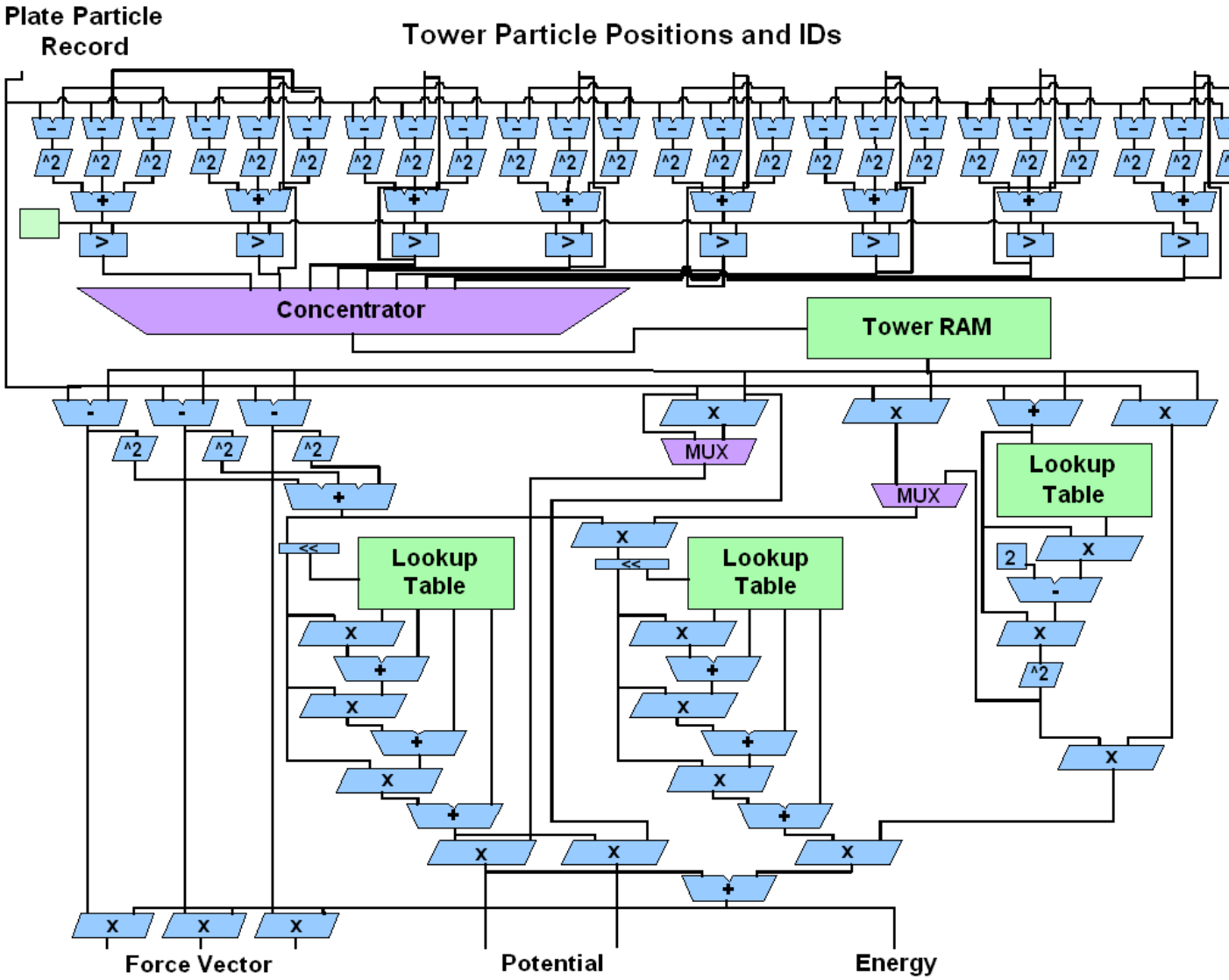
Specialized hardware (with tables, parameters) where:

- Inner loop
- Simple, regular algorithmic structure
- Unlikely to change

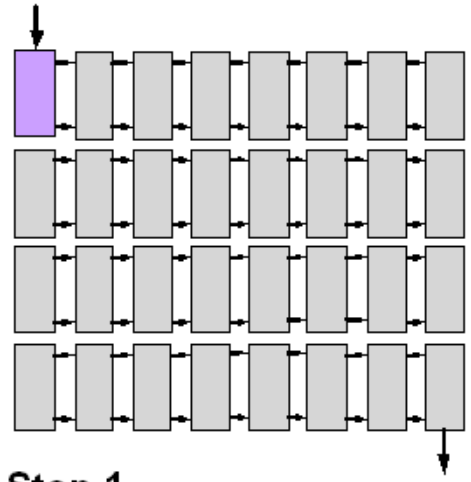
Examples:

- Electrostatic forces
- Van der Waals interactions

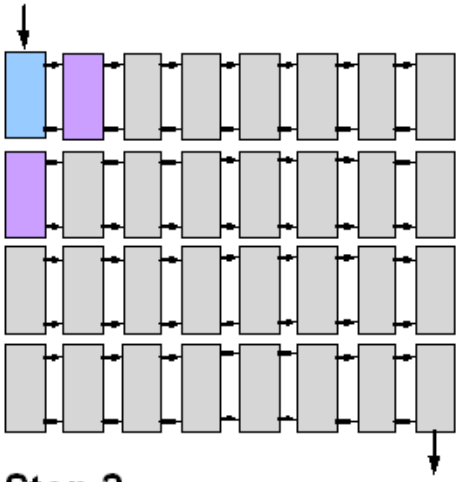
Example: Particle Interaction Pipeline (one of 32)



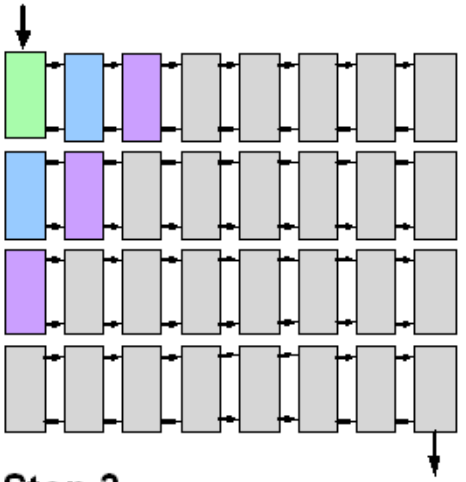
Array of 32 Particle Interaction Pipelines



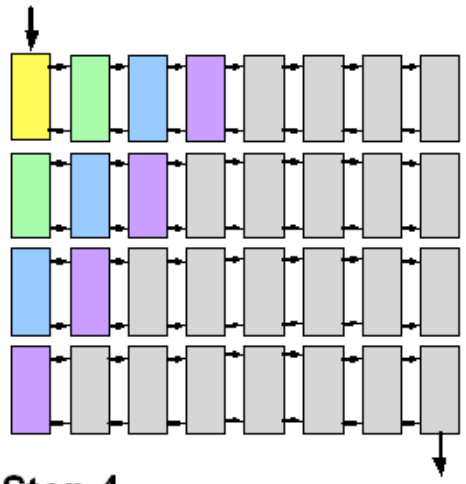
Step 1



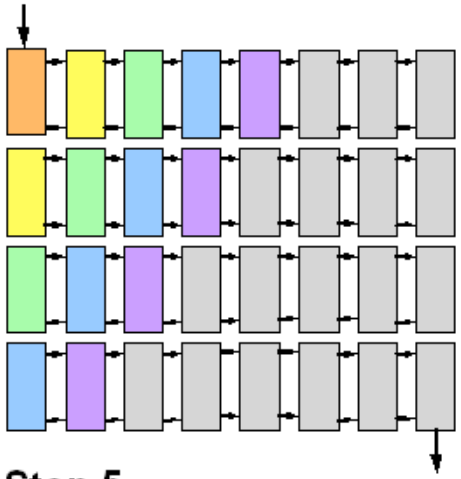
Step 2



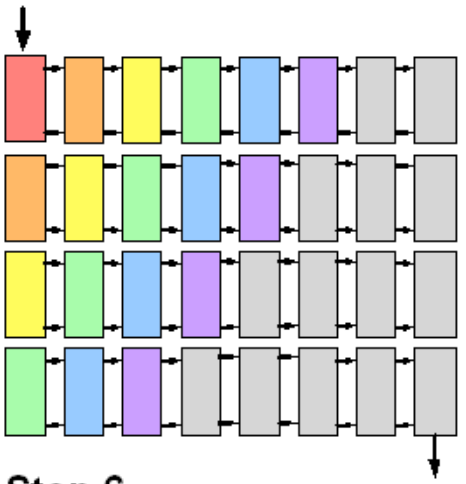
Step 3



Step 4



Step 5



Step 6

Advantages of Particle Interaction Pipelines

- Save area that would have been allocated to
  - _ Cache
  - _ Control logic
  - _ Wires
- Achieve extremely high arithmetic density
- Save time that would have been spent on
  - _ Cache misses,
  - _ Load/store instructions
  - _ Misc. data shuffling

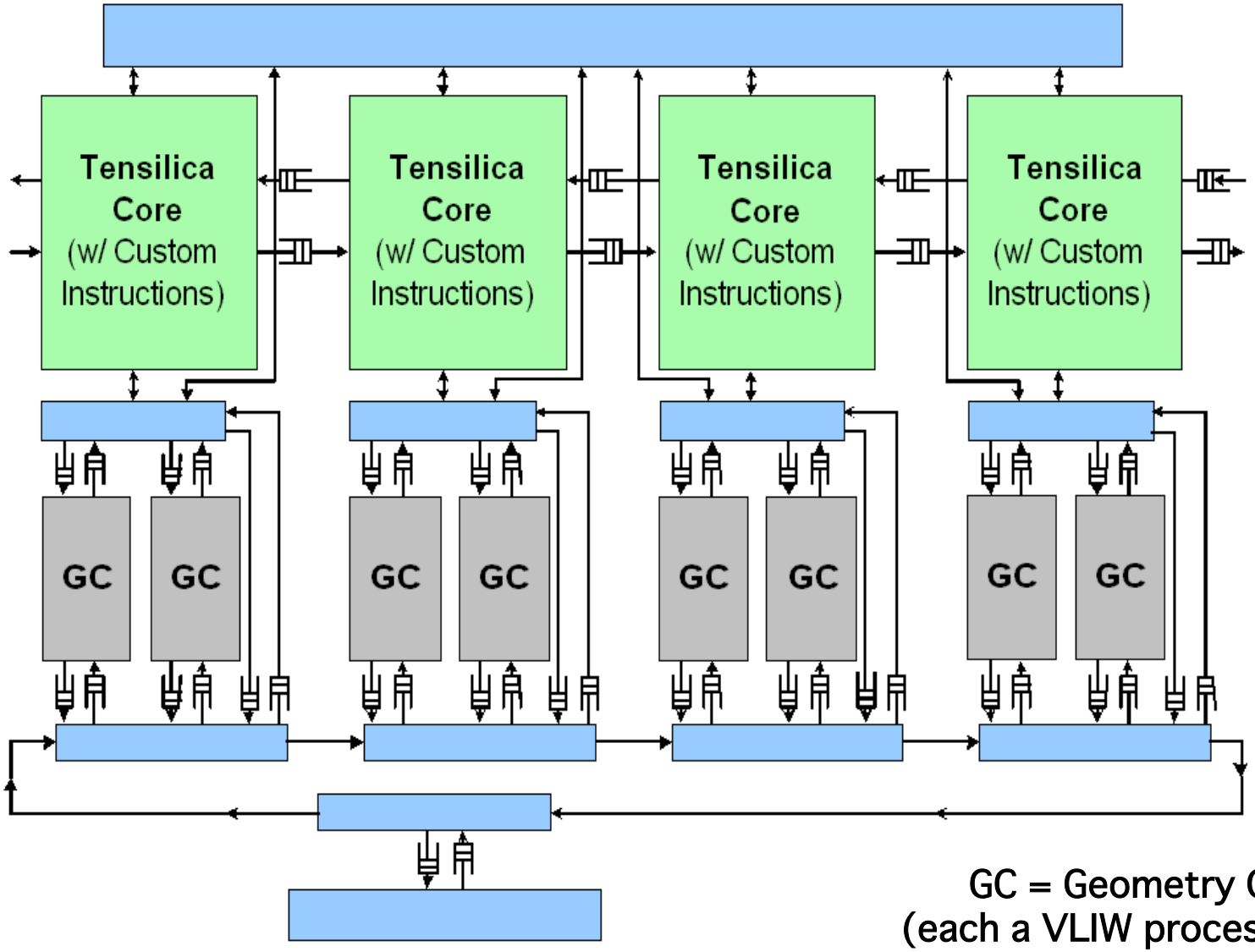
Where We Use Flexible Hardware

- Use programmable hardware where:
  - Algorithm less regular
  - Smaller % of total computation
    - E.g., local interactions (fewer of them)
  - More likely to change
- Examples:
  - Bonded interactions
  - Bond length constraints
  - Experimentation with
    - New, short-range force field terms
    - Alternative integration techniques

Forms of Parallelism in Flexible Subsystem

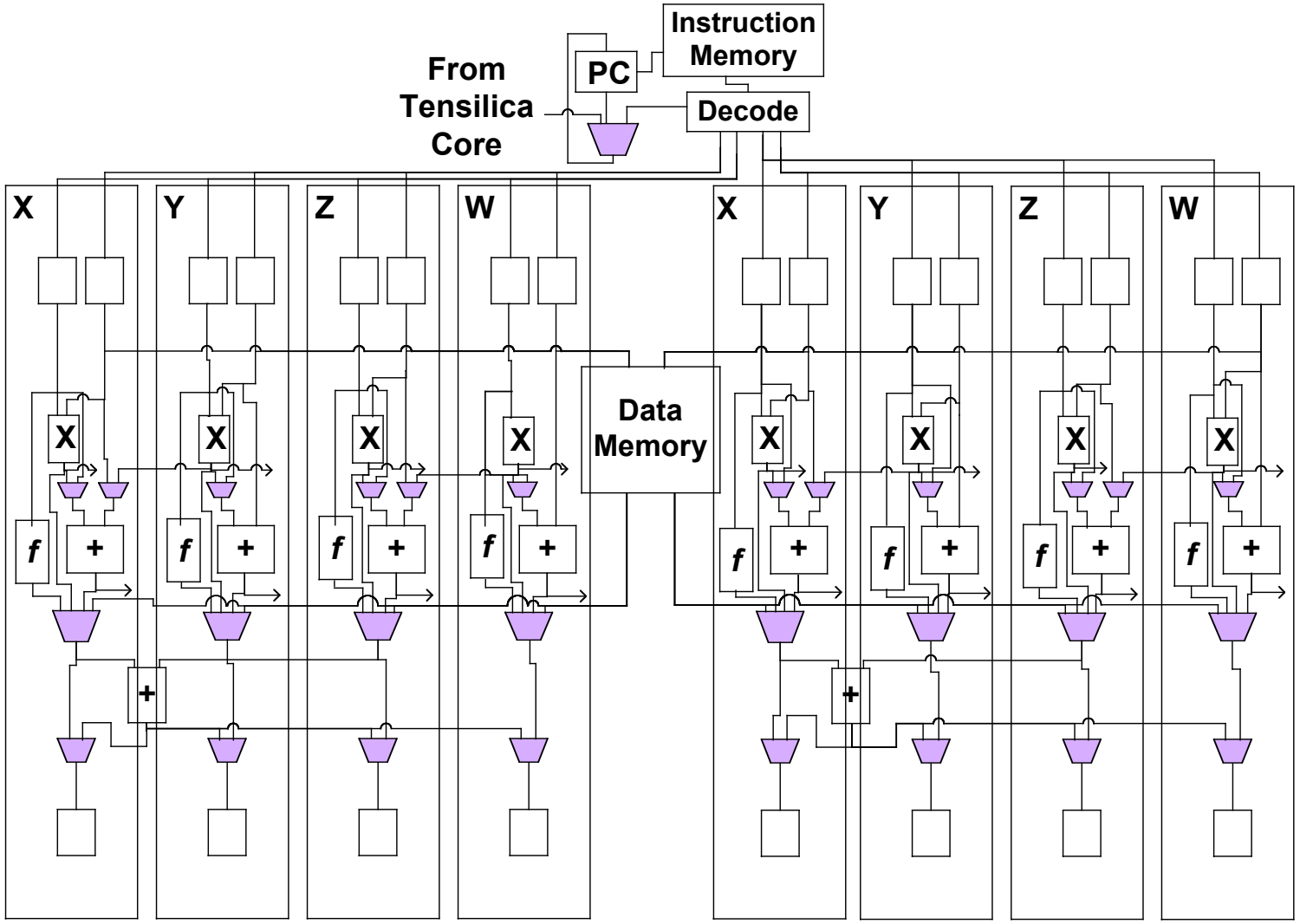
- The Flexible Subsystem exploits three forms of parallelism:
  - _ Multi-core parallelism (4 Tensilicas, 8 Geometry Cores)
  - _ Instruction-level parallelism
  - _ SIMD parallelism _ calculate on 3D and 4D vectors as single operation

Overview of the Flexible Subsystem



GC = Geometry Core
(each a VLIW processor)

Geometry Core
(one of 8; 64 pipelined lanes/chip)



But Communication is Still a Bottleneck

- Scalability limited by inter-chip communication
- To execute a *single* millisecond-scale simulation,
  - Need a huge number of processing elements
  - Must dramatically reduce amount of data transferred between these processing elements
- Can't do this without fundamentally new algorithms:
  - A family of *Neutral Territory (NT) methods* that reduce pair interaction communication load significantly
  - A new variant of Ewald distant method, *Gaussian Split Ewald (GSE)* which simplifies calculation and communication for distant interactions
  - *These are the subject of a different talk.*

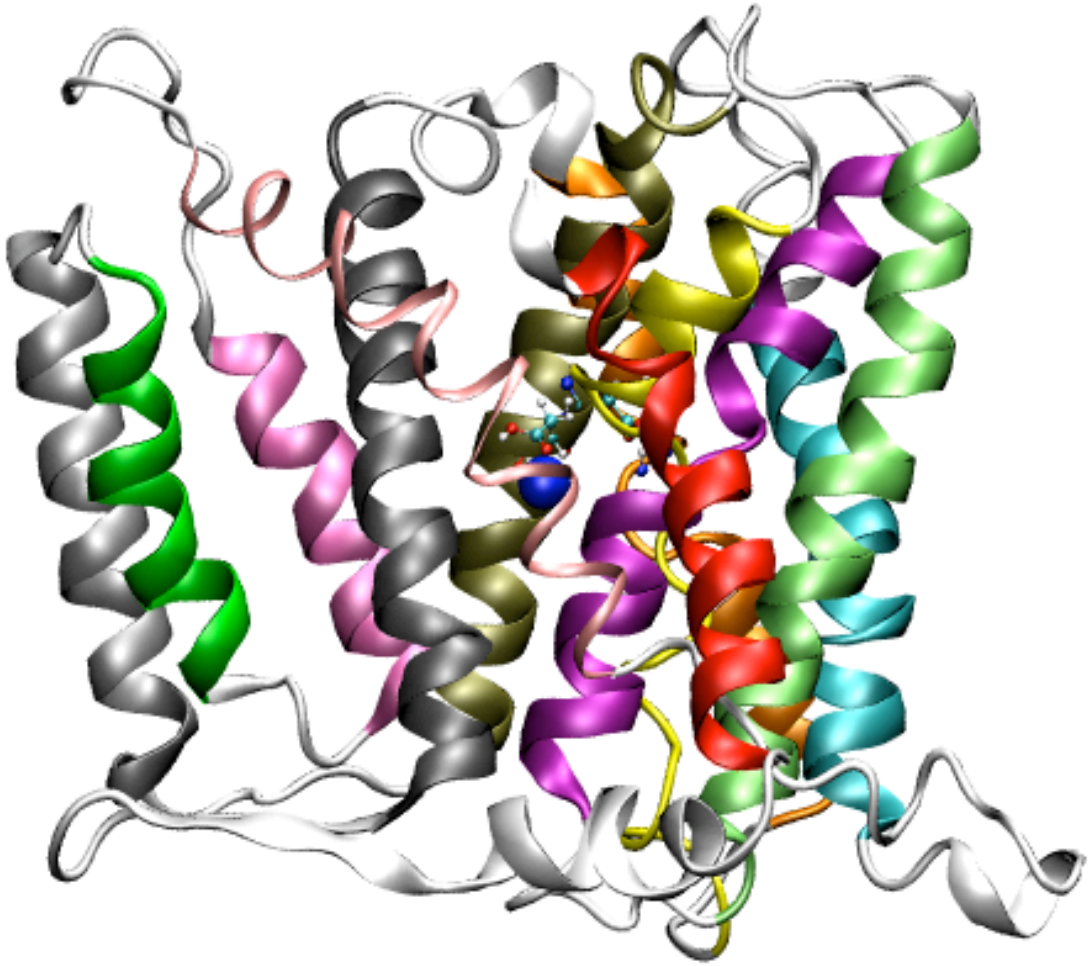
An Open Question That Keeps Us Awake at Night

Are Force Fields Accurate Enough?

- Nobody knows how accurate the force fields that everyone uses actually are
  - Can't simulate for long enough to know (*until we use Anton for the first time!*)
  - If problems surface, we should at least be able to
    - Figure out why
    - Take steps to fix them
- But we already know that fast, single MD simulations will prove sufficient to answer at least some major scientific questions

Example: Simulation of a Na+/H+ Antiporter

Cytoplasm



Periplasm

Our Functional Model of the Na+/H+ Antiporter